

How to draw combinatorial map? From graphs and edges to corner rotations and permutations

Dainis Zeps, Paulis Kikusts

Institute of Mathematics and Computer Science,
University of Latvia,
29 Rainis blvd., Riga, Latvia
dainize@mii.lu.lv
paulis@mii.lu.lv

Abstract. In this article we consider combinatorial maps approach to graphs on surfaces, and how between them can be establish terminological uniformity in favor of combinatorial maps in way rotations are set as base structural elements and all other notions are derived from them. We set this approach as rotational prevalence principle. We consider simplest way how to draw combinatorial map, and ask how this approach in form of rotational prevalence could be used in graphs drawing practice and wider in algorithms. We try to show in this paper that the use of corners in the place of halfedges is much more natural than that of halfedges. Formally there is no difference between both choices, but corner approach is much more clear and concise, thus we advocate for that.

1 Introduction

We deal in this article with some simple considerations and observations of how to draw combinatorial map, and how it comes in connection with traditional drawing of graphs, fig. 1, see fig. 4 in 3.2 too.

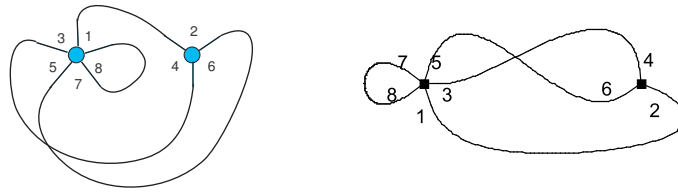


Fig. 1. Images of the same combinatorial map drawn in the plane.

To draw either graph or combinatorial map doesn't much differ if both require to come to picture of points and lines confining areas in the plane. Combinatorial map's approach suggest its own natural way how to come to its image and we try to follow directly this line. Moreover, we aim to build some base for graph's image what would follow from rotational approach, and, even more, establish it as sort of principle, calling it rotational prevalence when we go out from assumption that rotations in the graph defined as combinatorial map might serve as starting point for all other notions around the topology of the graph. We apply this approach in our case of building combinatorial map's drawer, accompanying with some questions what would be helpful for graph drawing environments.

Combinatorial map theory is new area of combinatorics which development may have considerable impact on topological graph theory. Besides, combinatorial functions around these maps are both theoretical tools and effectively calculable means that could be developed in complex environments for applications. These directions, as it seems to us, are weakly developed, partially maybe due to fact that combinatorial maps and apparatus around them as if repeat all what is already present in topological graph theory. But combinatorial maps as branch of combinatorics may be investigated independently from graph theory using e.g. such simple tools as permutations, and use whatever results already achieved in e.g. permutational group theory. Astonishing results of application of constellations (what is some generalization of combinatorial maps) in Riemann geometry [11] show that this area deserve to be researched with much more effort than before.

Combinatorial maps invention is attributed to Tutte [22,23], developed by Stahl [19,20,21] and many other researchers, see e.g. [12,1,13]. Combinatorial map approach is based on discoveries of rotations in graphs, see Heffter [6,7], Edmonds [2] and Jacques [8]. These facts are researched by many, see e.g. [5,24,16,14]. One way to introduce combinatorial map is by pair of permutations, see [1,11]. Lando and Zvonkin in their book "Graphs on Surface and Their Applications" [11] points at Jacques [8], who first used these constructions under the name of constellations. Constellations as generalization of combinatorial maps are used by Lando and Zvonkin in Riemann geometry [11], see [15] too. We do researches in these area since 1993 [26,9,27,28,29,30,31,32,34,35], that is based on permutational representation of rotations.

Specifically, combinatorial maps as purely combinatorial structure could deserve wider application in graph theory and some of its applications as e.g. graph drawing, where combinatorial maps could serve as more basic structure than graph itself. Rotations could be the combinatorial objects on which all other graph-theoretical invariants could be represented. In this article we try to reconstruct such option with questioning how to implement these observations more directly.

We tried to built combinatorial map drawing tool where we used directly combinatorial map approach to depict its eventual visual image, see fig. 4 in 3.3, in the same time giving account to the fact that all programming tradition used by graph drawings is based on rather far from rotational way of thinking

in combinatorics. The question we post in this article is - Can *rotations before edges and vertices* approach be applied directly in graph drawings particularly with some sensible outcome that could serve for applications?

By developing tools for combinatorial map drawing and by giving ways to the visualization of combinatorial maps we hope to widen the use of them in topological graph theory.

2 From graph on surface to set of rotations in the graph

For purposes of this paper we consider combinatorial map as arbitrary pair of permutations.

Combinatorial map as purely combinatorial object naturally models graph (in general hypergraph) embedded into oriented surfaces. By discovery of Heffter [6,7] and reminder of Edmonds [2] we know that vertex rotation, i.e., rotational order of edges around vertex, taken for all vertices, fixes graph on some orientable surface. But, if we fix rotational order of edges in faces too, than these two rotations, vertex rotation and face rotation, are sufficient to code all the graph, i.e., no sets of vertices and edges are to be specified, because these two rotations already cipher the graph on some orientable surface.

Moreover, coding these rotations with permutations and performing operations within purely permutational calculus, we may maintain all operations upon graph in the same way, i.e., using operations on permutations, that usually is done in vertices-edges-faces operational framework.

There is very natural way how to come from vertices and edges to permutational setting. Let graph $G = (V, E)$ be fixed by vertex rotation. If we depict each unoriented edge as pair of oriented edges in a way that outgoing edge comes first before incoming edge in rotation of edges around vertices in clockwise direction, we observe that oriented edges around faces are oriented all in one direction, and this direction is anticlockwise. See fig. 2 left. Now we are to do one more abstract operation – we replace the corner of the face that was formed by two oriented edges by halfedges, i.e., incoming head and outgoing tale, and call this new object *corner*. By the way, doing this, trivially, we replace double cover of borders of faces with corners of oriented edges with simple cover of corners of halfedges. See fig. 2.

These corners are as many as oriented edges in the graph, i.e., double as many as edges in the graph. Of course, these corners cover all oriented edges only once. Now, by fixing vertex and face rotations we actually fix two corner rotations, correspondingly around vertices and faces. The size of both rotations is the same, that of number of corners. Actually two more rotations of the same size appear, i.e., two edge rotations, which we get if we establish new adjacency of corners across what former was edge. Edge rotations in graphs have only 2-cycles, thus corresponding permutations are involutions. In case of hypergraphs edge rotations may have orbits of arbitrary size, i.e., corresponding to size of hyperedge.

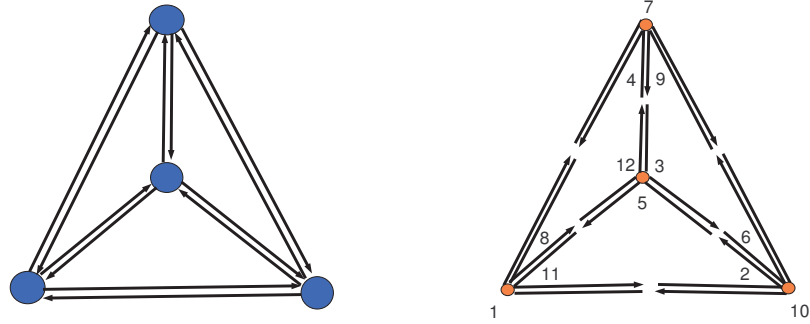


Fig. 2. Graphs K_4 embedding in plane fixed by rotation of alternatively outgoing and incoming edges around vertices, left, and by vertex rotation and face rotation of corners of edges, right. Numbering corners we come to permutations, which stand for rotations. Corners around vertices represent permutation $(18\bar{1})(26\bar{0})(35\bar{2})(479)$ and around faces permutation $(17\bar{0})(25\bar{1})(369)(48\bar{2})$. Two edge rotations are respectively $(12)(34)(56)(78)(9\bar{0})(\bar{1}\bar{2})$ and $(14)(23)(58)(67)(9\bar{2})(\bar{0}\bar{1})$.

Now, in case of graphs with possibly hyperedges we have four rotations of equal size that fix this graph on surface and without any necessary additional information, of course. But, four rotations are redundant. Both edge rotations have the same cycle structure, or *passport* [11]. Besides, taking only three rotations, but all in one direction, say, anticlockwise, multiplication of them is equal to identity permutation, [11]. (For that reason combinatorial map is 3-constellation.) This means that third permutation always may be calculated from the two given. Now, in case of graphs without hyperedges edge rotations are involutions without fixed points with respect to number of corners. If we fix edge rotation once and for ever, because its passport is constant, we may calculate all permutations from only one given permutation except two edge rotations which don't fix particular graph but some larger class. We further should see that this class is the class of the knot, to which this graph belongs.

2.1 From operations with graphs on surfaces to permutational calculus

With having corners (C) introduced we directly and naturally come to permutational calculus. Corners are as many as oriented edges in the graph, so number of corners $\|C\| (= m)$ is equal to $2\|E_G\|$, i.e., for graphs without hyperedges this number is always even. Thus, corners we may label (or equate at least for practical purposes) with natural numbers from interval $[1..m]$. We have four permutations corresponding to four rotations of the same size m . By the way, identical permutation of degree m as vertex rotation should have graph consisting from $m/2$ isolated edges in correspondence.

2.2 Hypergraphs and pairs of permutations

By the way, for graphs we should have only even permutations, but in case of hypergraphs arbitrary permutations come before. To fix hypergraph we need three permutations, or, by fixing edge rotation, both vertex and face rotation. In general, two permutations always have some hypergraph in correspondence [29]. In [29] we call combinatorial maps corresponding to hypergraphs *partial maps* in opposition to *graphical maps* for corresponding to graphs without hyperedges. The name of partial map is motivated by the fact that partial map may be considered as map with cut out some faces, thus being as if partial with respect of that graphical map with restituted cut out faces.

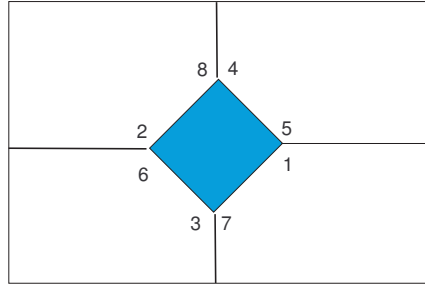


Fig. 3. Example of partial map with vertex rotation $P = (15)(26)(37)(48)$, face rotation $Q = (17452896)$ and edge rotation $R = (1423)(56)(78)$. One of edges, (1423) , is not graphical but hyperedge. See cover of two books [11] and [33].

See in fig. 3 example of hypergraph, graphs K_4 embedding on torus with cut out face. The cut out face stands for hyperedge of degree four. That we are to do with hypergraph we get from the fact that edge rotation is not graphical.

3 Graphical combinatorial maps introduced

Now we are ready to start combinatorial map calculus, which are based on permutational calculus, where permutations act on elements of set C , that stand for corners earlier introduced.

We multiply permutations from left to right. Graphical *combinatorial map* is pair of permutations, (P, Q) , with $P \cdot Q^{-1} (= \rho)$ being involution (i.e., orbits only of length two) without fixed elements. P and Q are correspondingly *vertex rotation* and *face rotation*. For graphical combinatorial map we distinguish two edge rotations, *inner edge rotation* $\pi = Q^{-1}P$ and *edge rotation* $\rho = P \cdot Q^{-1}$. We assume permutations acting on set of elements C , usually natural numbers from 1 to m , $m = \|C\|$. We try to work within set of maps with fixed π calling them *normalized maps*. Mostly we use one particular choice of π equal to $(12) \dots$

$(2k-1, 2k)$, $k \geq 1$. The same convention is used in [11]. Under these assumptions, graphical combinatorial map is characterized by one permutation or one rotation, say vertex rotation P .

3.1 One-one correspondence between permutations and graphs on surface

The fact we come to is worth to be appreciated specially. Two rotations fix combinatorial map, but under convention that inner edge rotation is fixed for some whole class of combinatorial maps it suffices only one permutation to specify graphical combinatorial map, see 3.5 lower. This feature is preserved under multiplication of permutations, as it is shown in [28]. As the result most of operations may be done within this convention.

Permutational calculus and theory around permutations usually coming under name of permutational group theory is very developed, e.g. [25]. What does it mean that graphs on surface one-one correspond to permutations? One way to perceive this is to state that whatever theorem in permutations work equally in this class of graphs. One of our aims of this article is to turn attention on how to maintain this permutational calculus available in graph theoretical applications, thus giving eventual access to whatever useful in theory around permutations.

However, there may appear applications where we need to go outside the class of normalized maps, but it is easy to bookkeep these situations, but use, whenever possible, benefits of normalized maps.

3.2 Drawing graph on surface

A graph with loops and multiedges on an orientable surface corresponds to arbitrary graphical combinatorial map in very natural way. One intuitively well based way to persuade oneself about this is to draw directly this graph in the plane in the following way. Let first put as many points in the plane as orbits in the vertex rotation with *edge-ends* clockwise around with corners following in cyclical order for each corresponding point, see fig.1, a. Further, let us unite two edge-ends with corners (following clockwise) a and $b(= a^\pi)$. I.e., corners that form orbit of inner edge rotation π are to be united with an edge in the drawing, thus justifying an option to speak about *edges of combinatorial maps*. It is easy to see that changing orientation from clockwise to anticlockwise we had to use edge rotation ρ in place of inner edge rotation π . Thus, two opposite directions of rotations give two distinct edge rotations both in drawings and combinatorially.

See e.g. fig. 4 where combinatorial map $P = (18753)(264)$ is drawn. In a) we place points in the plane corresponding to orbits in vertex rotation. In b) to d) to get drawing of combinatorial map, we are to unite pairs of edge-ends $(2k-1, 2k)$ with curves for edges from $k = 1$ to 4. In d) we have the drawing fulfilled. Notice that loops correspond to edge-ends that go out from the same point in the plane.

Further examples of combinatorial map drawings performed with programmed map drawer see in section 4.2.

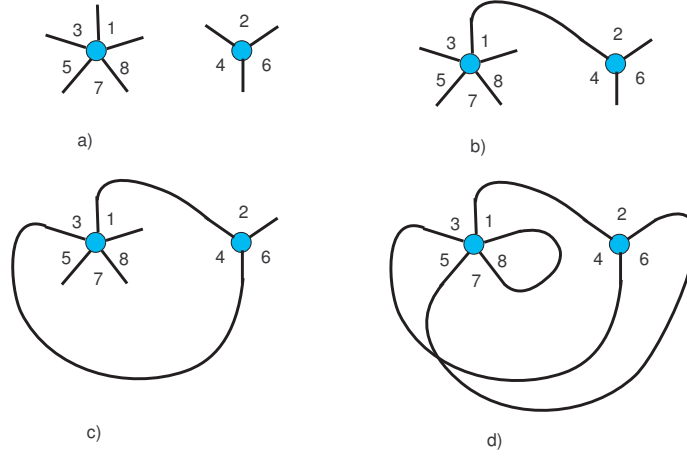


Fig. 4. a) Putting two points in the plane with edge-ends and corresponding corners from orbits of the given permutation $(18753)(264)$. b) Uniting edge-ends with corners 1 and 2. Notice the order edge-end and the corresponding corner in the clockwise direction. c) Uniting edge-ends 3 and 4. d) Uniting two left edge-end pairs. Notice that last "edge" of the map is loop.

3.3 Definition of edge of combinatorial map

Higher we started to speak about *edge of combinatorial map* in some informal, intuitively based way.

Thus, we are motivated to consider the edge of combinatorial map in some more formally way.

Let Ω_p and ω_p be correspondingly set of orbits and some orbit of permutation p .

Definition 1 Let for a given map P, Q, π, ρ sextet $e = \langle \omega_P^1, \omega_P^2; a, b, c, d \rangle$ be such that $a, b \in \omega_P^1$, $c, d \in \omega_P^2$, $(ac) \in \Omega_\pi$, $(bd) \in \Omega_\rho$ and $(ac)^P = (bd)$. We call e edge of combinatorial map.

Let us use some denotations for transitive permutations in cyclical form. For one corner orbit (a) we use small letter, i.e., a , for arbitrary length orbit (or part of orbit) we use capital letter, i.e., A might be orbit or its part (a_1, \dots, a_k) , $k \geq 0$. In this manner aA denotes some transitive permutation p in cyclical form with corner a being distinguished and rest part being A .

Using this convention we may write an edge of a map in the form

$$\langle baA, dcC; a(=c^\pi), b(=d^\rho), c(=a^\pi), d(=b^\rho) \rangle.$$

Let us fix this as a lemma.

Lemma 2 *Orbits (ac) and (bd) correspondingly of inner edge rotation and edge rotation have edge*

$$\langle baA, dcC; a(=c^\pi), b(=d^\rho), c(=a^\pi), d(=b^\rho) \rangle$$

of combinatorial map in correspondence with corresponding equalities being held.

We must observe that the edge we speak about turns actually into a loop in case orbits baA and dcC coincide. But corners a, b, c, d all should be distinct nevertheless except in cases of isolated or hanging edges or loops. So isolated edges or loops has two distinct corners, $\langle a, b; a, a, b, b \rangle$ being isolated edge, and $\langle ab, ab; a, b, b, a \rangle$ being isolated loop. Hanging edge or loop has three distinct corners, e.g., $\langle ba, c; a, b, c, c \rangle$ being hanging edge, and $\langle abc, cab; a, b, c, b \rangle$ being hanging loop.

E.g., for normalized map $P = (18753)(264)$ $e = \langle (18753), (264); 1, 3, 2, 4 \rangle$ is an edge, comp. fig. 4. In case we don't want to specify rotations from where edges go out we use shorter form, quartet notation only for four corners, e.g., we write for edge $e = \langle 1, 3, 2, 4 \rangle$.

3.4 Graphs on surfaces

Usually treating graphs on surface we start from graph $G = (V, E)$ and then supply it with cyclical succession of neighboring edges, i.e., with some permutation, say, Π that works on set of unoriented edges. It turns out that graph on surface might be considered as simpler structure than graph supplied with rotations, i.e., rotations of vertices and faces already fix all graph-on-surface structure without taking much concern on what kind on surface "graph" was "embedded".

This may be adequately taken into account by fixing graph on surface by *corners between neighboring edges* in rotations. Let graph G be fixed by edge rotations around vertices. Let C be set of corners between edges and two rotations (or actually permutations) are given: rotation of corners around vertices R_V and rotation of corners around faces R_F . It is easy to see that now some pairs of corners (c_1, c_2) are *connected via edge* in the way that $c_1^{R_V^{R_F^{-1}}} = c_2$. This fact is sufficient and crucial to establish one-one correspondence between graphs on surface and combinatorial maps.

Lemma 3 *For a graph G on surface with given rotations of corners of neighboring edges around vertices and faces R_V and R_F the pair of permutations (R_V, R_F) is graphical combinatorial map.*

Proof. To prove this assertion it suffices to show that multiplication $R_V \cdot R_F^{-1}$ is involution without fixed points. But for arbitrary corner x its *edge-adjacent* corner $x^{R_V^{R_F^{-1}}} = y$ is some other corner at some vertex and applying backwards we get that edge-adjacent corner to y is x , i.e., $y^{R_V^{R_F^{-1}}} = x$. But this means that multiplication $R_V \cdot R_F^{-1}$ is involution without fixed points.

To conclude we state that *edge of combinatorial map* (R_V, R_F) may be interpreted as edge of graph G on surface fixed by this pair of rotations (R_V, R_F) .

Corollary 4 *For combinatorial map (R_V, R_F) corresponding to graph G on surface sextet*

$$\langle baA = \omega_{V_G}, dcC = \omega_{V_F}, a = c^{R_V^{-1}R_F}, b = d^{R_V^{R_F^{-1}}}, c = a^{R_V^{-1}R_F}, d = b^{R_V^{R_F^{-1}}} \rangle$$

is map's edge that has edge (b, d) uniting corners b and d of graph G on surface in correspondence.

3.5 Combinatorial map as combinatorial structure of graph on surface

We may establish one-one correspondence between graphs on surfaces and graphical combinatorial maps. To get it we are to take set of corners C that should serve as set of elements of combinatorial maps. Further, we fix as arbitrary graph on surface pair of rotations, for vertices and faces correspondingly, say, $R_v (= P)$ and $R_f (= Q)$. Now, inner edge rotation should be $\pi = P^{-1} \cdot Q$ and edge rotation $\rho = P \cdot Q^{-1}$. Now, combinatorial map base assumptions says that graph on surface is well defined whenever edge rotation ρ is involution without fixed points. This means that under these assumptions we have established one-one correspondence between graphs on surfaces and combinatorial maps.

In [28] we saw that we may fix one of edge rotation, say, inner edge rotation and work, whenever possible, within this constraint. Rightly, we may get along with one edge rotation, second edge rotation actually as variable being superfluous. We chose to fix inner edge rotation, and, by the way, in a very convenient way, fixing it $(12)(34)(56)\dots(2m-12m)$. Such maps we call normalized maps. Further, in [35] we see that we may go on with normalization and normalize knot in the map too. Then corresponding graphs on surfaces behave in complete correspondence with maps and we don't lose one-one correspondence between maps and graphs on surfaces.

All this taking into account we are not to be too specific to specify always what we have in mind when we speak, say, about corners, or some else aspect, either corresponding to maps or graphs on surfaces: we always use map-graph-on-surface correspondence.

3.6 Zig-zag walk and the knot

Let us perform zig-zag walk in the graph on surface. Taking corner c let us go on with applying π and ρ alternatively and we are to return to corner c , see [35]. I.e., this routine is cyclical. If we haven't exhausted all corners we take some free one and go on. In this way we get another rotation for a graph on surface, or map, that we call zig-zag walk for graphs on surface, and knot for maps. However, this rotation is not unique, i.e., it has 2^k possible values for k orbits in

the knot. It is easy to see that each change of orientation of some orbit of knot gives another well defined value of knot.

If we fix one particular value of knot we actually divide set of corners C into two equal size parts C_1 and C_2 so that we may express this particular knot as $\mu = \begin{cases} C_1 : \pi \\ C_2 : \rho \end{cases}$, where the expression $\mu = \begin{cases} C_1 : p \\ C_2 : q \end{cases}$ say, that within set C_1 we apply p , and within C_2 we apply q .

Let us color corners into two colors according their belonging either to C_1 or C_2 . Usually we use for C_1 green color and for C_2 red color.

Particular choice of knot μ partitions π into $\pi_1 \cdot \pi_2$, where we call π_1 *cut edges* and π_2 *cycle edges*. For *cut edge* $\langle a, b, c, d \rangle$ corners a, b receive one color and c, d another color. Correspondingly, for *cycle edge* $\langle a, b, c, d \rangle$ corners a, d receive one color and b, c another color. Thus we have that $P \cdot \pi_1 : C_1 \mapsto C_2$ or $P \cdot \pi_1 : C_2 \mapsto C_1$, i.e., changes color, and $P \cdot \pi_2 : C_1 \mapsto C_1$ or $P \cdot \pi_2 : C_2 \mapsto C_2$, i.e., retains color. In [28] was shown that by fixing μ map P may be expressed as multiplication $\gamma_1 \cdot \gamma_2 \cdot \pi_2$, where γ_1 acts within C_1 and γ_2 acts within C_2 . From this we get formula for knot μ . $\mu = \gamma_2 \cdot \pi \cdot \gamma_1^{-1}$, see [35]. Finally, in general we get

$$\mu = \begin{cases} C_1 : \pi \\ C_2 : \rho \end{cases} = \begin{cases} C_2 : \gamma_2 \pi \gamma_1^{-1} \\ C_1 : \gamma_1 \rho \gamma_2^{-1} \end{cases}.$$

One might ask what is knot for graph on surface? In [35] we show that actually the equivalence class of zig-zag walks is graph's on surface invariant. To come closer to the knot and what it means we may observe that partial map (P, μ) has edge rotation Q , i.e., this partial map, which face rotation is μ , is hypergraph with edges these of faces of original map. This fact says that fixing graph G on surface fixes another hypergraph with the same vertex rotation but hyperdegrees these of faces of G , in which case faces of this hypergraph should be knot of the graph.

3.7 Knotting of the map

Fixing knot μ for map P we get by multiplying permutations new rotation for maps α , i.e.,

$$P = \mu \cdot \alpha.$$

We call this rotation α knotting of map because of its peculiar features. Knotting belongs to selfconjugate class of maps, i.e., $\alpha^\pi = \alpha$, which form a group against multiplication[28].

Knotting α is responsible for edge structure of maps[35]. To see this we define knotting's symmetric form $A = \alpha \cdot \pi_1 = \gamma_1 \cdot \gamma_1^{-1}$, involution $\delta = \pi^{\gamma_1}$ and *edge structuring knot* $\varepsilon = \begin{cases} C_1 : \pi \\ C_2 : \delta \end{cases}$. In [35] we show that $\varepsilon^2 \cong A$, i.e., one of square roots of knotting's symmetric form is equal to edge structuring knot.

3.8 Normalizing knots

In [28] we considered class of maps with fixed knot. Actually this class has fixed both inner edge rotation π and edge rotation ρ . Such class may be called class of maps with normalized knot. We add one more restriction: normalized knot should have corners in augmented order.

In fig. 8 we see map with normalized knot. Odd corners have received one color and even corners another. But to get this feature we don't need to knot be normalized. It suffices for it to be only *partially normalized*, i.e., all edge should appear in the knot in the same order, that is, for all edges its less valued corner should come before, or reversely. In [35] we show for knot and edge structuring knot that if one of them is normalized then other is partially normalized.

3.9 Combinatorial adjacency of corners

By drawing graph we use vertices, edges and faces. Replacing these three types of objects with rotations we retain minimally necessary for what is the picture of the graph in the essence. One more way to see this clearly is to notice that combinatorial maps may be treated as adjacency relations of corners of halfedges of three types, vertex, face and edge adjacencies, see fig.5. Taking corners of two halfedges we may assemble them into rotation in three ways, see fig. 6: a) around common apex of corners, forming vertex rotation, b) by adjoining halfedges in way forming face rotation, and c) by flipping one corner to its mirror image and adjoining halfedges in way forming an edge rotation. In this we may replace vertices, edges and faces with adjacencies of corners of three types, and do just the same in replacing graphs with rotations in combinatorial maps. Let us name these three adjacencies correspondingly p-adjacency, q-adjacency and r-adjacency.

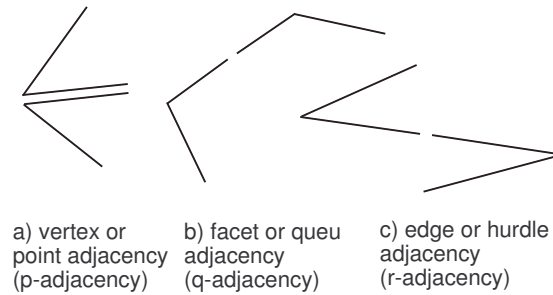


Fig. 5. Combinatorial adjacency of corners.

Let us say that map is fixed by p-adjacency and q-adjacency if rotations P and Q are given. Thus, normalized maps are fixed by p-adjacency or q-adjacency,

r-adjacency being fixed for whole class of maps. Knot appear to us as clockwise anticlockwise alternation of r-adjacency, or *left-right r-adjacency*. Edge structuring knot is as much as possible right along forwarded edges alternation of r-adjacencies. But this last in its name fixed behavior of edge structuring knot reveals itself only in case it is at least partially normalized.

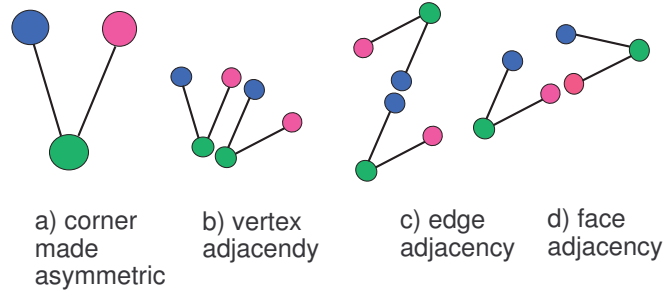


Fig. 6. Combinatorial adjacency of corners symmetrized with asymmetric corner. Asymmetry in corner made by marking vertex or red vertex, edge or blue vertex and face or red vertex. Uniting in rotations green, blue and red vertices we get correspondingly vertex, edge and face adjacency. Notice that in face adjacency one corner is flipped by mirror reflection.

In [11] similar result is achieved via canonical triangulation of faces of the graph and getting three involutions, see Construction 1.5.20 on page 49, [11]. We may do the same by making our corner asymmetric by labeling it with three vertices of different color, apex with green label, calling it vertex's vertex, left to apex with blue label, calling it edge vertex, and right to apex with red label, calling it face vertex. Now we make three color rotations, green rotation for vertex, blue rotation for edges and red rotation for faces. Now we have come to complete symmetry, what complies with symmetry in rotations in combinatorial map.

By the way, all considerations in this chapter persuades us that the choice to number corners in the graph picture to get combinatorial map is as legitimate operation as replacing the drawing of the graph with combinatorial map.

3.10 Constellations as combinatorial structure of Riemann surfaces [11,15]

Rotational approach gives direct generalization from Euclidean plane to ramified Riemann sphere [11,15] where we are to imagine graph being drawn on oriented surface X of arbitrary genus that serves as domain of ramified covering f of sphere S^2 , $f : X \rightarrow Y$.

How to use it here, and what it could mean for us here? This says that rotational structures are more basic and before choice of where and how we are to perform our picture's structure itself. We are as if before choice either to draw one nontrivial pictures on trivial surface, or many trivial pictures on sheets of nontrivial surface. But maybe, for technical reasons, we may get used to something in the middle, i.e., to divide global picture in smaller ones each of which we put on separate sheets. The use of rotational approach is quite insensible of where we are going to choose these middle points. But this might be only one of perspectives of rotational prevalence. Thus, in projecting whatever around rotational system we are to place them in the base, both rotations, operations with them and all apparatus around them, and only after all the rest.

4 On rotational prevalence

All before was told with the aim to show that rotations and permutations may be basic units of some combinatorial structures. In this order we would like to speak about rotational prevalence that would suggest to use these rotational aspects whenever and however necessary according their prevalence in theoretical settings.

4.1 Drawing maps

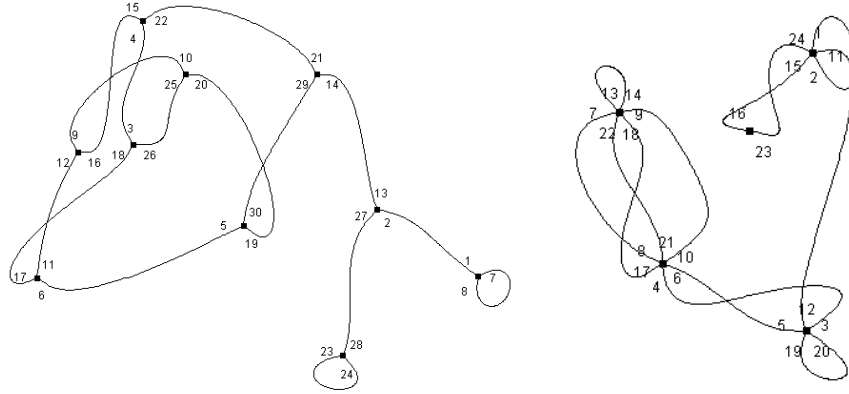


Fig. 7. Two examples of randomly generated maps. Drawings may make maps look beautiful, as we see!

If one used to combinatorial maps comes to someone graph drawing specialist with request to draw a map the latter asks the former to prepare graphical code for his/her combinatorial map. Doesn't it occur to him/her that combinatorial map is a simpler structure than the code of the graph to be drawn? This situation is paradigmatic, and requests to be cured somehow, or at least to ask how to teach graph drawing community to learn more about rotational systems in order to make rotational prevalence we speak about in this article as working principle.

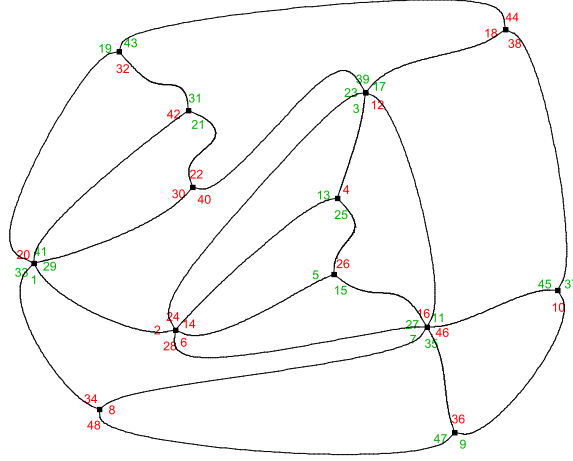


Fig. 8. Example of combinatorial map with normalized knot. Cover of set of corners with green and red cycles clearly seen. Corresponding graph is a 4-critical multiwheel[36].

What would be that that first had to teach to the second in case one needs to draw a graph? One way is to work according the schema given in this article, i.e., code the graph as two rotations of corners and calculate edges of the corresponding map. It would be formal implementation of rotational prevalence. But it wouldn't give much if we came to necessity to start to implement rotations as points, lines and bordered areas in the plane, that would correspond to the elements of the pictures of the graphs we are used to. Would these last objects to be redesigned in terms of some rotational geometry that with such easy could be done in case of Riemann geometry? In graph's rotational geometry classes of vertices, edges and faces don't differ, that is not the case for their opposites - points, lines and areas.

On the other hand, if we follow the line how rotational prevalence would work on level of theoretical considerations, we might keep as close as possible to it, and turn our attention on how to develop this ground level of rotational

calculus. Possibility to visualize maps and especially partial maps should be very useful in these efforts.

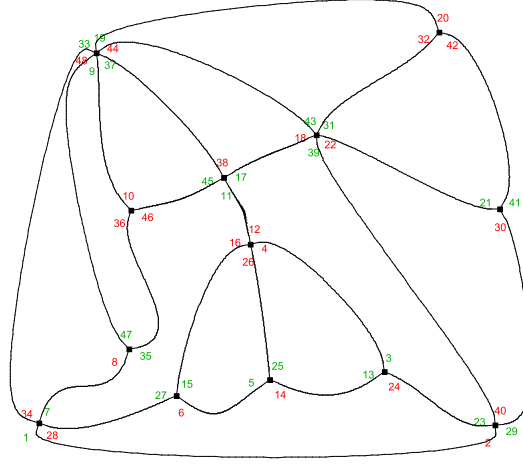


Fig. 9. Dual map of previous example.

4.2 Combinatorial map drawer used in this article

We have endeavored to build a small drawer for combinatorial maps using the simplest way by placing points supplied with edge-ends with corners for vertices in the plane and uniting them by edges from edge rotation according 3.3. The edge ends first was united by rectangular lines, which afterwards were smoothed into curve-like edges. The drawer was supplied with manual operation to move vertex in the plane, that was supported with corrector of all rest edge lines with respect to this relocated vertex.

The drawer used in this article was built by Paulis Kikusts. The experience used here was typical for graph drawing area, see e.g., [10,3,4]. See pictures of combinatorial maps and partial maps in figures 7, 8, 9, 10, 11 and 12.

In fig. 10 examples of simplest partial maps are given. In fig. 8 an example of *4-critical multigraph* from [36] implemented planar and coded as map is drawn. Here, map is normalized with respect to its knot, and green and red cycles that cover corner set are clearly seen. In fig. 7 drawing examples of two arbitrary generated maps are seen.

Similarly as in rotational case we are to choose two notions from triple - points, lines and areas - to be varied independently. Tradition says that we leave areas to be determined by first two. Thus, our rotational prevalence doesn't

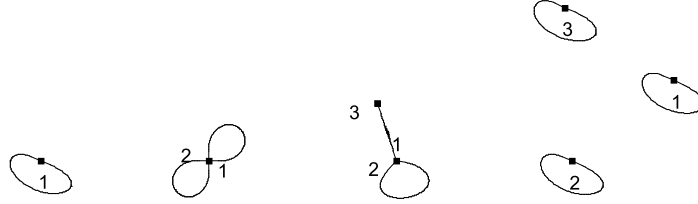


Fig. 10. Four examples of small partial maps: 1) (id_1, id_1) ; 2) $((12), (12))$; 3) $((12), (123))$; 4) (id_3, id_3)

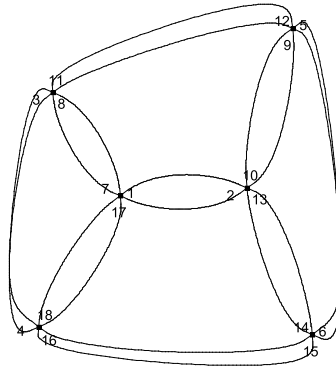


Fig. 11. Prism map (P) with cut out faces as example of partial map (P, π) .

exceed this border of point-line-area geometry. The break into this area would need new ideas. By the way, on simple way of uniforming these three geometrical quantities in a geometry of areas is suggested by cubic combinatorial maps in the approach of C.H.C. Little, see [12,1].

4.3 Combinatorial maps as combinatorial structure of algorithms and data structures

One way to widen the application of rotational relations would be to remember that rotations work as sort of theorems starting from rotational data structures, cycles, graphs, constellations, and going on to operations and rotational theorems, and more daring theoretical principles, as in the case of Riemann geometry. It may find applications in how we build computer algorithms and before all the

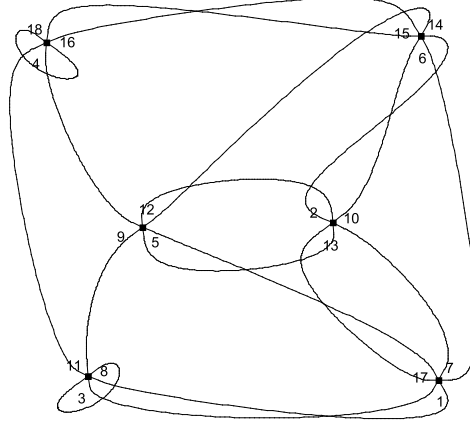


Fig. 12. Partial map with two hyperedges of size 5 and 13 correspondingly defined by pair of permutations $(177)(20\bar{3})(3\bar{1}8)(4\bar{8}6)(6\bar{5}4)(59\bar{2})$ and $(138)(2\bar{2}4)(4\bar{1}9)(\bar{5}6\bar{8})(5\bar{3}7)(76\bar{0})$.

graph algorithms and maybe starting with implementing permutational calculus itself. First author of this article tried to use such philosophy in programming of a graph algorithm with but some practical outcome in somewhat good organized program.

However, rotational prevalence need be supported with theoretical investigations. We need to develop combinatorial map theory itself. Application of partial maps particularly might be useful there. Because of this possibility to visualize maps and especially partial maps would be very useful in these efforts.

5 Conclusions

Combinatorial map approach as purely combinatorial may serve as good example for graph theory and applications that combinatorial approaches may be used not only as theoretical support but as direct operational tools as well. By building operational outset of some systems combinatorial base may be used not only as required invariants of the elements of system, but the base for its structure itself already on level of its design and implementation. In this approach we suggest to use rotations-before-edges-faces-and-vertices approach as some rotational prevalence principle. We ask how we could apply this rotational prevalence in real conditions of graph drawing systems.

6 Acknowledgements

We thank doctoral students of second author Rihards Opmanis and Rudolfs Opmanis for their help in building map drawer program.

References

1. Bonington Paul, Little Charles, *The Foundations of Topological Graph Theory*, Springer, 1995, 178pp.
2. Edmonds J. K. *A combinatorial representation for polyhedral surfaces*, Notices Amer. Math. Soc. (1960), 646.
3. Freivalds K., Kikusts P. *Optimum layout adjustment supporting ordering constraints in graph-like diagram drawing*, Proc. of the Latvian Academy of Sciences, Section B, Vol. 55, No 1, 2001, pp. 43-51.
4. Freivalds K., Dogrusoz U., Kikusts P. *Disconnected graph layout and the polyomino packing approach*, Proc. of Symposium Graph Drawing 2001, Lecture Notes in Computer Science, vol. 2265, 2002, pp. 378-391.
5. Giblin P. J. *Graphs, Surfaces, and Homology*. John Willey & Sons, 1977.
6. Heffter L. *Über das Problem der Nachbargebiete*. Math. Ann., 1891, vol. 38, 477-508.
7. Heffter L. *Über metacyklische Gruppen und Nachbarconfigurationen*, Math. Ann. 50, 261- 268, 1898.
8. Jacques A. *Constellations et graphes topologiques*, - In: "Combinatorial Theory and its Applications: (P.Erdos, A.Renyi, V Sos eds.)", North-Holland, Amsterdam, 1970, vol. II, 657-673.
9. Kikusts P., Zeps D. *Graphs on surfaces*, Conf. LMS, Riga, 1994.
10. Kikusts P., Ručevskis P. *Layout algorithms of graphlike diagrams for GRADE Windows graphic editors*, Proc. of Symposium Graph Drawing '95, Lecture Notes in Computer Science, vol. 1027, 1996, pp.361364.
11. Lando Sergei K., Zvonkin Alexander K. *Graphs on Surfaces and Their Applications*, Encyclopaedia of Mathematical Sciences, Vol 141, Low Dimensional Topology II, Springer, 2004, 455pp.
12. Little C.H.C. *Cubic combinatorial maps*, J.Combin.Theory Ser. B 44 (1988), 44-63.
13. Liu Yanpei A. *Polyhedral Theory on Graphs*, Acta Mathematica Sinica, New Series, 1994, Vol.10,No.2, pp.136-142.
14. Mohar Bojan, Thomassen Carsten. *Graphs on Surfaces*, J. Hopkins Univ. Press, 2001.
15. Panov D., Zvonkine D. *Counting meromorphic functions with critical points of large multiplicities*, zapiski nauchnich seminarov POMI, Tom 292, 2002, 28pp.
16. Ringel G. *Map Color Theorem*, Springer Verlag, 1974.
17. Rosenstiel P., Read R.C. *On the principal edge tripartition of a graph*, Discrete Math. 3,195-226, 1978.
18. Stahl S. *The Embedding of a Graph - A Survey*. J.Graph Th., Vol 2 (1978), 275-298.
19. Stahl S. *Permutation-partition pairs: A combinatorial generalisation of graph embedding*, Trans Amer. Math. Soc. 1 (259) (1980), 129-145.
20. Stahl S. *A combinatorial analog of the Jordan curve theorem*, J.Combin.Theory Ser.B 35 (1983), 28-38.
21. Stahl S. *A duality for permutations*, Discrete Math. 71 (1988), 257-271.

22. W.T. Tutte. *What is a map? New directions in the theory of graphs*. Academic Press, NY, 1973, pp. 309-325.
23. Tutte W.T. *Combinatorial maps*, in *Graph theory*, chapter X, 1984.
24. White A. T. *Graphs, Groups and Surfaces*, North-Holland, P.C., 1973.
25. Wielandt H. *Finite Permutations Groups*, Academic Press, New York, 1964.
26. Zeps D. *Graphs with rotations as selfdual facet systems*, unpublished, 1993, 6pp.
27. Zeps D. *Graphs with rotation in permutation technique*, KAM Series, N 94-274, 1994, Prague, Charles University, 8pp.
28. Zeps D. *Graphs as Rotations*, KAM-Series, (96-327), Prague, 1996, 11pp, arXiv:0909.0104v1.
29. Zeps D. *Graphs with rotations:partial maps*, KAM Series, 97-365, 1996, 12pp.
30. Zeps D. *The use of the combinatorial map theory in graph-topological computations*, KAM Series, 97-364, Prague, 1997, 8pp.
31. Zeps D. *The theory of Combinatorial Maps and its Use in the Graph-topological computations*, PhD thesis, 1998.
32. Zeps D. *Using combinatorial maps in graph-topological computations*, KAM Series, 99-438, Prague, 1997, 11pp.
33. Zeps D. *Combinatorial Maps. Tutorial*, Online book, vixra.org/abs/1001.0030, 2004.
34. Zeps D. *Combinatorial map as multiplication of combinatorial knots*, KAM Series, (2008-864), Prague, 2008, 7pp, hal-00408153.
35. Zeps D. *Combinatorial Maps with Normalized Knot*, arXiv:1010.2615v1, 2010, 14pp.
36. Zeps D. *On building 4-critical plane and projective plane multiwheels from odd wheels*, arXiv:1202.4862v1, 2012, 12pp.